Functional Maps: A Flexible Representation of Maps Between Shapes

Maks Ovsjanikov[†] Mirela Ben-Chen[‡] Justin Solomon[‡] Adrian Butscher[‡] Leonidas Guibas[‡]

[†] LIX, École Polytechnique [‡] Stanford University

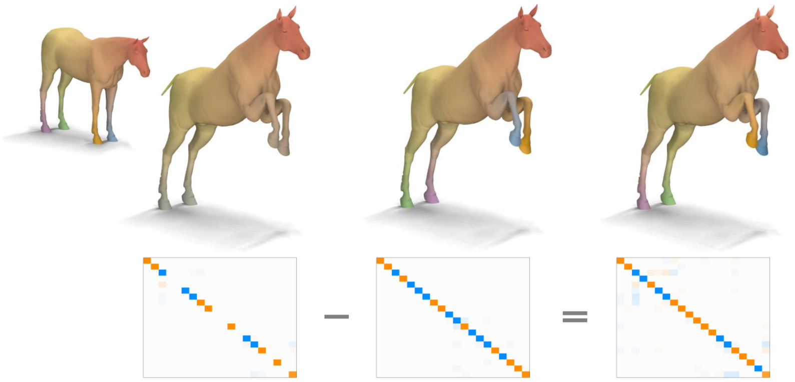


Figure 1: *Horse algebra: the functional representation and map inference algorithm allow us to go beyond point-to-point maps. The source shape (top left corner) was mapped to the target shape (left) by posing descriptor-based functional constraints which do not disambiguate symmetries (i.e. without landmark constraints). By further adding correspondence constraints, we obtain a near isometric map which reverses orientation, mapping left to right (center). The representation allows for algebraic operations on shape maps, so we can subtract this map from the ambivalent map, to retrieve the orientation preserving near-isometry (right). Each column shows the first 20x20 block of the functional map representation (bottom), and the action of the map by transferring colors from the source shape to the target shape (top).*

Abstract

We present a novel representation of maps between pairs of shapes that allows for efficient inference and manipulation. Key to our approach is a generalization of the notion of map that puts in correspondence real-valued functions rather than points on the shapes. By choosing a multi-scale basis for the function space on each shape, such as the eigenfunctions of its Laplace-Beltrami operator, we obtain a representation of a map that is very compact, yet fully suitable for global inference. Perhaps more remarkably, most natural constraints on a map, such as descriptor preservation, landmark correspondences, part preservation and operator commutativity become linear in this formulation. Moreover, the representation naturally supports certain algebraic operations such as map sum, difference and composition, and enables a number of applications, such as function or annotation transfer without establishing point-to-point correspondences. We exploit these properties to devise an efficient shape matching method, at the core of which is a single linear solve. The new method achieves state-of-the-art results on an isometric shape matching benchmark. We also show how this representation can be used to improve the quality of maps produced by existing shape matching methods, and illustrate its usefulness in segmentation transfer and joint analysis of shape collections.

Keywords: shape matching, representation, correspondence.

Links: DL PDF Web

1 Introduction

Shape matching lies at the core of many operations in geometry processing. While several solutions to rigid matching are well established, non-rigid shape matching remains difficult even when the space of deformations is limited to e.g. approximate isometries. Part of the difficulty in devising a robust and efficient non-rigid shape matching method is that unlike the rigid case, where the deformation can be represented compactly as a rotation and translation, non-rigid shape matchings are most frequently represented as pairings (correspondences) of points or regions on the two shapes. This representation makes direct map estimation and inference intractable, since the space of possible point correspondences is exponential in size. For example, isometric matching techniques try to find correspondences that preserve geodesic distances as well as possible, but such optimization problems can be shown to be an NP-hard subclass of the quadratic assignment problem [Çela 1998]. Perhaps more importantly, this representation does not naturally support constraints such as map continuity or global consistency.

Additionally, in many practical situations, it is neither possible nor necessary to establish point-to-point correspondences between a pair of shapes, because of inherent shape ambiguities or because the user may only be interested in approximate alignment. Such ambiguous or approximate map inference is difficult to phrase in terms of point-to-point correspondences.

The majority of existing methods try to tackle these challenges by limiting their search for correspondences between a small set of landmark points and extending those to a dense set of correspondences on entire shapes during final post-processing ([Bronstein et al. 2006; Huang et al. 2008; Lipman and Funkhouser 2009; Kin-Chung Au et al. 2010; Ovsjanikov et al. 2010; Kim et al. 2011; Tevs et al. 2011; Sahillioğlu and Yemez 2011] among many others). This strategy has also been justified theoretically, since under general conditions a small set of landmark correspondences is known to be sufficient to obtain a unique dense mapping between isometric surfaces ([Lipman and Funkhouser 2009; Ovsjanikov et al. 2010]).

Nevertheless, although this landmark-based approach reduces the complexity of the solution space it still relies on representing shape maps as point-to-point correspondences, making it difficult to incorporate global constraints or return meaningful results when establishing point correspondences is not possible due to the presence of only coarse similarities or symmetry ambiguities.

In this paper we present a novel approach for inference and manipulation of maps between shapes that tries to resolve the issues above in a fundamentally different way. Rather than putting in correspondence points on the shapes, we propose to consider mappings between *functions* defined on the shapes. This notion of correspondence generalizes the standard point-to-point map since every pointwise correspondence induces a mapping between function spaces, while the opposite is, in general, not true. However, this generalized representation is: 1) flexible, since it allows choosing a basis for the function space on each shape and representing the mapping as a change of basis matrix and 2) well-suited for shape-matching, since many natural constraints on the map become *linear* constraints on the functional map. As we show in the rest of this paper, our representation works especially well when combined with the eigenfunctions of the Laplace-Beltrami operator, by benefiting from their multi-scale, "geometry-aware" nature. This allows us, in particular, to devise a simple algorithm that achieves state-of-the art results on an isometric shape matching benchmark and at the heart of which is a single linear solve. We also demonstrate the usefulness of this representation on a number of tasks including improving existing maps, segmentation transfer and joint analysis of shape collections without establishing point-to-point correspondences.

2 Related Work

Both shape matching in general and non-rigid shape matching in particular are relatively well-established fields with several recent books (e.g. [Bronstein et al. 2008]) and surveys [van Kaick et al. 2011b] dedicated exclusively to this subject. It is out of scope for the current article to cover the breadth of all existing shape matching methods; we concentrate on the various classes of underlying *representations* for maps between pairs of shapes and indicate ways in which they are optimized for in the literature.

As mentioned in the introduction, the vast majority of existing shape matching methods represent a map between a pair of shapes as a point-to-point correspondence. Since it is infeasible to optimize over such correspondences directly, most methods aim to obtain a sparse set of point correspondences and extend them to dense mappings [Bronstein et al. 2006; Huang et al. 2008; Lipman and Funkhouser 2009; Kin-Chung Au et al. 2010; Ovsjanikov et al. 2010; Kim et al. 2011; Tevs et al. 2011; Sahillioğlu and Yemez 2011]. Because sparse point correspondences are inherently discrete, common ways to enforce global consistency include preservation of various quantities between pairs or sets of points, including geodesic distances [Bronstein et al. 2006; Huang et al. 2008; Sahillioğlu and Yemez 2011], various spectral quantities [Jain et al. 2007; Mateus et al. 2008; Sharma and Horaud 2010; Ovsjanikov et al. 2010], or embedding shapes into into canonical domains [Lipman and Funkhouser 2009] based on landmark correspondences, or a combination of multiple geometric and topological tests [Dubrovina and Kimmel 2011; Kin-Chung Au et al. 2010].

A related set of techniques aims to establish shape part or segment correspondences rather than reliable point-to-point matches, e.g. [Golovinskiy and Funkhouser 2009; Xu et al. 2010; Pokrass et al. 2011; Huang et al. 2011; van Kaick et al. 2011a]. Such techniques either pre-segment the shape and try to establish part correspondences, or more recently phrase the segmentation and correspondence (and possibly labelling) in a joint optimization framework [Kalogerakis et al. 2010; Pokrass et al. 2011; Huang et al. 2011] which generally avoids the need to establish reliable pointwise correspondences. In this paper we show how segment correspondences can be used as constraints to establish high quality point matches.

Finally, some methods optimize the deformation of one shape to align it with another, rather than optimizing the correspondences directly [Zhang et al. 2008; Yeh et al. 2010]. In the majority of cases, however, such methods still rely on point correspondences either during alignment or pre-processing as feature matches.

We note that some recent methods have concentrated on measuring and optimizing consistency of *sets* of maps [Nguyen et al. 2011; Kim et al. 2011] and showed superior performance to optimizing individual correspondences. These applications show the importance of algebraic operations on maps (averages, differences), which are challenging to do in the point-to-point correspondence domain.

Our use of spectral quantities is also closely related to spectral embeddings [Rustamov 2007] and their application in shape matching [Jain et al. 2007; Mateus et al. 2008; Ovsjanikov et al. 2008]. However, unlike such methods our framework does not assume one-to-one correspondences between eigenfunctions of the Laplace-Beltrami operator. This difference is crucial for both removing the combinatorial complexity present in these methods (e.g. sign ambiguities, order switching) and achieving superior results in practice.

One common characteristic of all existing non-rigid shape matching methods is that they lead to difficult, non-convex, non-linear optimization problems. In this paper, we argue that this is primarily because maps between shapes are represented as point or segment correspondences, making it inherently difficult to devise map inference methods using global constraints. On the other hand, we show that by generalizing the notion of a map to include pairings of functions instead of points, map inference can be phrased as a linear system of equations. One danger of this generalization is that the solution may not correspond to a point-to-point map. We show simple regularization techniques that help avoid this possibility.

Note that analyzing mappings through their effect on function spaces is a common theme used in various fields of mathematics. A famous example can be found in the field of Representation Theory (see for instance [Weyl 1946]). Here, one relates the different ways in which a compact Lie group (a continuous group of transformations, e.g. the group of rotations of plane) can act on itself to the induced *linear* action of the group on functions.

3 Contributions

The key contribution of this paper is a new representation for maps between pairs of shapes as linear transformations between the corresponding function spaces. We show how this notion of a map generalizes the standard point-to-point representation and yet has the following key advantages:

- By using the Laplace-Beltrami basis for the function space on each shape, the map can be well-approximated using a small number of basis functions and expressed simply as a matrix.
- Most natural constraints on maps, such as descriptor preservation, landmark correspondences, part preservation and operator commutativity become linear in the functional formulation, enabling extremely efficient inference.
- Maps in this representation can be manipulated via standard algebraic operations e.g. addition, subtraction, composition.

Last but not least, functional maps can be useful even when they do not correspond to point-to-point maps for information or attribute transfer between shapes, shape collection analysis, and other shape processing tasks.

4 Functional Map Representation

To set the stage for functional mappings as a generalization of classical point-to-point mappings, let $T : M \to N$ be a bijective mapping between manifolds M and N (either continuous or discrete). Then, T induces a natural transformation of derived quantities, such as functions on M. To be precise, if we are given a scalar function $f : M \to \mathbb{R}$ then we obtain a corresponding function $g : N \to \mathbb{R}$ by composition, as in $g = f \circ T^{-1}$. Let us denote this induced transformation by $T_F : \mathcal{F}(M, \mathbb{R}) \to \mathcal{F}(N, \mathbb{R})$, where we use $\mathcal{F}(\cdot, \mathbb{R})$ to denote a generic space of real-valued functions. We call T_F the *functional representation* of the mapping T. We now make the following two simple remarks:

Remark 4.1. *The original mapping T can be recovered from T_F.*

Indeed, to recover the image $T(a)$ of any point a on M, construct an indicator function $f : M \to \mathbb{R}$, s.t. $f(a) = 1$ and $f(x) = 0 \;\forall\, x \neq a \in M$. By construction if $g = T_F(f)$, then $g(y) = f \circ T^{-1}(y) = 0$ whenever $T^{-1}(y) \neq a$ and 1 otherwise. Since T is assumed to be invertible, there is a unique point y s.t. $T(a) = y$. Thus, g must be an indicator function of $T(a)$ and $T(a)$ is the unique point $y \in N$ s.t. $g(y) = 1$.

Remark 4.2. *For any fixed bijective map $T : M \to N$, T_F is a* linear *map between function spaces.*

To see this, note $T_F(\alpha_1 f_1 + \alpha_2 f_2) = (\alpha_1 f_1 + \alpha_2 f_2) \circ T^{-1} = \alpha_1 f_1 \circ T^{-1} + \alpha_2 f_2 \circ T^{-1} = \alpha_1 T_F(f_1) + \alpha_2 T_F(f_2)$.

We may paraphrase these remarks to say that knowledge of T_F is *equivalent* to knowledge of T. And while T may be a complicated mapping between surfaces, T_F acts linearly between function spaces.

Now suppose that the function space of M is equipped with a basis so that any function $f : M \to \mathbb{R}$ can be represented as a linear combination of basis functions $f = \sum_i a_i \phi_i^M$. Then,

$$T_F(f) = T_F\left(\sum_i a_i \phi_i^M\right) = \sum_i a_i T_F\left(\phi_i^M\right) .$$

In addition, if N is equipped with a set of basis functions $\{\phi_j^N\}$, then $T_F\left(\phi_i^M\right) = \sum_j c_{ij}\phi_j^N$ for some $\{c_{ij}\}$ and

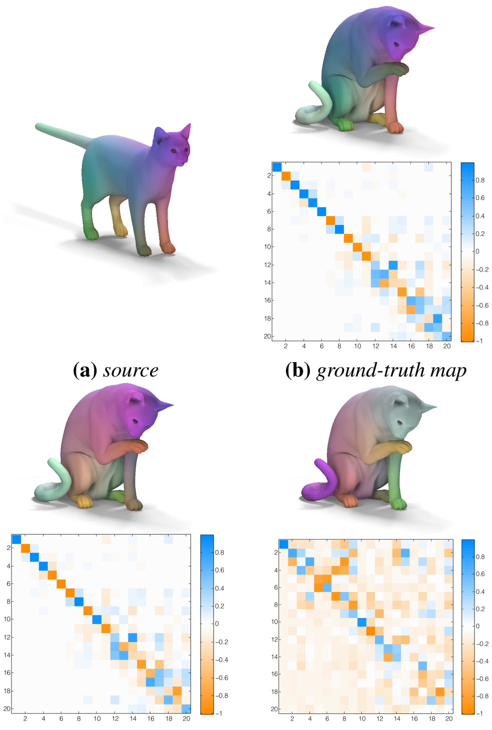
$$T_F(f) = \sum_i a_i \sum_j c_{ij}\phi_j^N = \sum_j \sum_i a_i c_{ij}\phi_j^N . \tag{1}$$

Therefore if we represent f as a vector of coefficients $\mathbf{a} = (a_0, a_1, ...a_i, ...)$ and $g = T_F(f)$ as a vector $\mathbf{b} = (b_0, b_1, ..., b_i, ...)$, then Eq. 1 simply says: $b_j = \sum_i a_i c_{ij}$, where c_{ij} is independent of f and is completely determined by the bases and the map T. In particular c_{ij} is the j^{th} coefficient of $T_F(\phi_i^M)$ in the basis $\{\phi_j^N\}$. Note that C has a particularly simple representation if the basis functions $\{\phi_i^N\}$ are orthonormal with respect to some inner product $\langle\cdot,\cdot\rangle$, namely $c_{ij} = \langle T_F(\phi_i^M), \phi_j^N\rangle$.

We conclude with the following key observation:

Remark 4.3. *The map T_F can be represented as a (possibly infinite) matrix C s.t. for any function f represented as a vector of coefficients* $\mathbf{a}$ *then* $T_F(\mathbf{a}) = C\mathbf{a}$.

This remark in combination with the previous two remarks shows that the matrix C fully encodes the original map T.



(a) *source* **(b)** *ground-truth map*

(c) *left to right map* **(d)** *head to tail map*

Figure 2: *Two shapes with three maps between them, each rendered as a point-to-point mapping through color correspondence (top) and its functional representation (bottom) with colors proportional to matrix values. Note that the least isometric map in (d) leads to a less sparse functional matrix.*

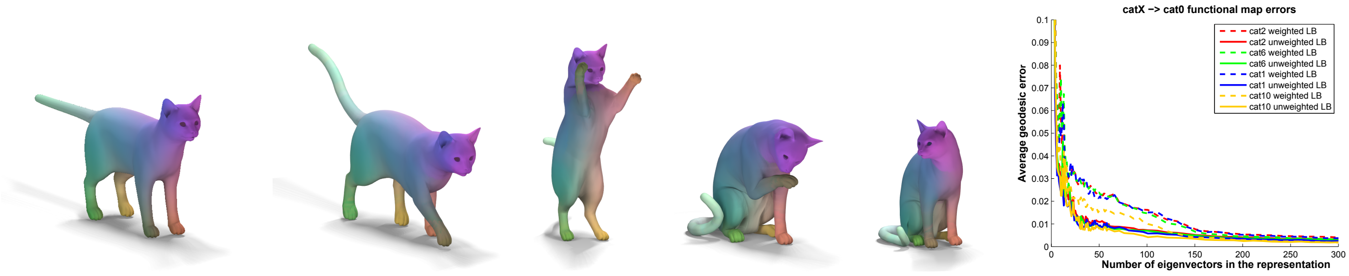
Motivated by this discussion, we now turn towards the definition of *linear functional mappings* that are strictly more general than functional representations of classical point-to-point mappings. The point of view that we take is to downplay the mapping T and focus our attention on the matrix C. We thus define:

Definition 1. *Let $\{\phi_i^M\}$ and $\{\phi_j^N\}$ be bases for $\mathcal{F}(M, \mathbb{R})$ and $\mathcal{F}(N, \mathbb{R})$, respectively. A generalized linear functional mapping $T_F : \mathcal{F}(M, \mathbb{R}) \to \mathcal{F}(N, \mathbb{R})$ with respect to these bases is the operator defined by*

$$T_F\left(\sum_i a_i \phi_i^M\right) = \sum_j \sum_i a_i c_{ij} \phi_j^N ,$$

where c_{ij} is a possibly infinite matrix of real coefficients (subject to conditions that guarantee convergence of the sums above).

Example. As an example, consider a pair of shapes in Figure 2 with three bijective maps between then: two approximate isometries (the ground-truth map and the left-right mirror symmetric map) and one map that puts the head and tail in correspondence. For each map, the point-to-point representation is shown as color correspondence while the functional representation is shown as a heat map of the matrix $C_{0..20\times 0..20}$, where we used the Laplace-Beltrami eigenfunctions as the basis for the function space on each shape. Note that the functional representations of the near-isometric maps are close to being sparse and diagonally dominant, whereas the representation of the map that associates the head with the tail is not. Also note that none of the maps is diagonal, an assumption made by previous algorithms [Jain et al. 2007; Mateus et al. 2008; Ovsjanikov et al. 2008].



(a) *target* **(b)** *Cat10* **(c)** *Cat1* **(d)** *Cat2* **(e)** *Cat6* **(f)** *errors*

Figure 3: *Average mapping error vs. number of eigenvalues used in the representation. For each shape with a known ground-truth point-to-point correspondence (shown as a color correspondence), we computed its functional representation and measured its accuracy in finding a point-to-point map. Note that although more eigenvalues lead to an increase in accuracy, maps that correspond to bigger deformations require more basis vectors, capturing the intuition that near-isometric maps are more compactly represented.*

5 Functional Representation Properties

As we have noted above, the functional representation of a pointwise bijection can be used to recover its representation as a correspondence, and is thus equivalent. Note, however, that this does not imply that the space of bijections coincides with the space of linear maps between function spaces, as the latter may include functional mappings not associated with any point-to-point correspondence.

Perhaps the simplest example of this is a functional map D that maps every function on one shape to the constant 0 function on the other — D clearly cannot be associated with any pointwise correspondences since all such functional maps must, by definition, preserve the set of values of each function. Nevertheless, by going to this richer space of correspondences, we obtain a representation that has several key properties making it more suitable for manipulation and inference.

Intuitively, functional maps are easy to manipulate because they can be represented as matrices and thus can benefit from standard linear algebra techniques. To make this intuition practical, however, the size of the matrices must be moderate (i.e. independent of the number of points on the shapes), and furthermore map inference should be phrased in terms of linear constraints in this representation. In the following sections we will show how to achieve these goals first by choosing the appropriate basis for the function space on each shape (Section 5.1) and then by showing how many natural constraints on the map can be phrased as linear constraints on the functional map (Section 5.3), reducing shape matching to a moderately-sized system of linear equations (Section 6).

5.1 Choice of basis

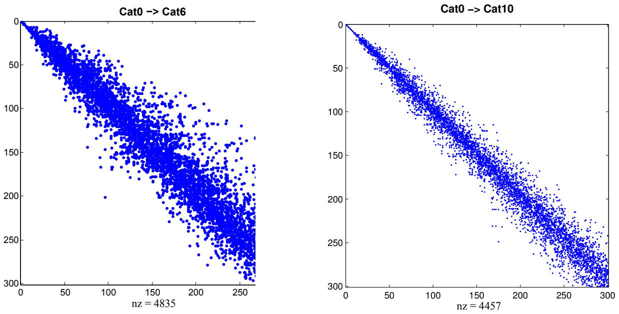
As noted above, the functional map representation is *flexible* in the sense that it gives us the freedom to choose the basis functions for the functional spaces of M and N. Indeed, if we choose the basis functions to be indicator functions at the vertices of the shapes, then C is simply the permutation matrix which corresponds to the original mapping. However, other choices of bases are possible, which can lead to significant reductions in representation complexity and are much better suited for *continuous* mappings between shapes — the desired behavior in the majority of practical applications.

Perhaps the two most important characteristics for choosing a basis for functional maps are compactness and stability. Compactness means that most natural functions on a shape should be well approximated by using a small number of basis elements, while stability means that the *space* of functions spanned by all linear combinations of basis functions must be stable under small shape deformations. These two properties together ensure that we can represent the action T_F using a small and robust subset of basis functions and we need only consider a finite *submatrix* $C_{0..m\times 0...n}$, for some moderate values of m and n, of the infinite matrix C (Definition 1). In other words, for a given function f, represented as a vector of coefficients $\mathbf{a} = (a_0, a_1, ...a_i, ...)$, we would like $\sum_j \sum_i a_i c_{ij} \phi_j^N \approx \sum_{j=0}^{n} \sum_{i=0}^{m} a_i c_{ij} \phi_j^N$, for some fixed small values of m and n.

In this paper, we will concentrate on shapes undergoing near-isometric deformations, for which we will use the first n Laplace-Beltrami eigenfunctions as the basis for their functional representations (where $n = 100$ throughout all of our experiments, independent of the number of points on the shape). This choice of basis is natural, since eigenfunctions of the Laplace-Beltrami operator are ordered from "low frequency" to "higher frequency," meaning that they provide a natural multi-scale way to approximate functions, and as a result functional mappings, between shapes. Moreover, although individual eigenfunctions are known to be unstable under perturbations, suffering from well-known phenomena such as sign flipping and eigenfunction order changes, the *space* of functions spanned by the first n eigenfunctions of the Laplace-Beltrami operator can be shown to be stable under near-isometries as long as the n^{th} and $(n+1)^{\text{st}}$ eigenvalues are well separated, as shown for example in the work of [Kato 1995].

To illustrate the role of the choice of basis on the functional representation, we compare two widely-used discretizations of the Laplace-Beltrami operator and measure their ability to capture a ground-truth point-to-point correspondence using a fixed number n of basis functions. In particular, we consider, the cotangent weight scheme of Meyer et al. [2002] with and without area normalization (in the latter case, each vertex is assigned a uniform weight, while in the former the weight is proportional to the sum of the areas of triangles around the point). Figure 3 shows the average error induced by the functional representation for a set of pairs of deformed versions of the cat shape provided in the TOSCA [Bronstein et al. 2008] dataset. Each of these shapes contains 27.8K points, with a known ground-truth correspondence. We represented this pointwise correspondence between the cat0 shape and the others using an increasing number of eigenvectors, and for each point x computed its image as: $T(x) = \arg\max_y T_F(f)(y)$ where f is the indicator function at the point x on shape M. I.e. $T(x)$ is the point where the function $g = T_F(f)$ is maximal. The error is measured in average geodesic error units (see [Kim et al. 2011] Figure 7 middle for illustration).

Note that the eigenfunctions of the unweighted discretization of Laplace-Beltrami operator provide a more compact representation



(a) *cat6* **(b)** *cat10*

Figure 4: *Sparsity pattern of the matrices C corresponding to two out of 4 maps shown in Figure 3. Only cells where $|C| > 0.11$ are shown. Note that more than 94% are not. Note also that the functional matrix for the more deformed shape cat6 is also farther from being diagonal.*

of the map for the same quality of reconstruction. This may be because this discretization is less sensitive to volume distortion (and only sensitive to non-conformal distortions). Moreover, note that only $30 - 40$ eigenfunctions are sufficient to represent the ground truth map to a quality that is extremely close to the ground-truth point-to-point map. Since a functional representation with 40 eigenvectors implies a matrix of size 40×40, this means that even without exploiting its sparsity (described below), this representation has 1600 values, which is a nearly 17 times memory savings over a permutation of size 27.8K.

Sparsity. In addition to the multi-scale property of the functional representation with the Laplace-Beltrami eigenfunctions, we also point out that near-isometric maps induce matrices C that are nearly sparse and thus can be stored efficiently. Indeed, if the shapes M and N are isometric and T is an isometry, it is easy to see that the matrix C_{ij} can be non-zero only if ϕ_j^M and ϕ_i^N correspond to the same eigenvalue. In particular, if all eigenvalues are non-repeating, C is a diagonal matrix. In practice, we observe that if T is only approximately an isometry, the matrix C is still close to being sparse, or funnel-shaped. Figure 4 shows the sparsity patterns of the matrices C corresponding to two of the maps shown in Figure 3. In particular, note that over 94% of the values of these matrices are below 0.1. Let us stress, however, that the functional matrix C stops being diagonal very quickly under even mild non-isometric deformations, and this effect is especially pronounced for high-frequency eigenfunctions (Figure 2 illustrates the same effect). While this poses fundamental challenges to previous spectral methods [Jain et al. 2007; Mateus et al. 2008; Ovsjanikov et al. 2008], the functional representation naturally encodes such changes.

5.2 Continuity

Another major advantage of using the functional representation of the mapping is that it naturally handles map continuity unlike the point-to-point or segment-to-segment bijection which is inherently discrete. Here continuity means three distinct phenomena:

Continuity under changes of the input function. This means that the image of a function $T_F(f) = C\mathbf{a}$ varies continuously under changes of the vector of coefficients $\mathbf{a}$ and thus under the changes of the function for a fixed mapping C. This property is useful since in most natural settings the desired mapping is continuous.

Continuity of the image function. The Laplace-Beltrami operator is inherently well-suited for representing *smooth* functions on the shapes. Thus, for any fixed number n, and any set of coefficients $\mathbf{a}$, the function $f = \sum_{i=0}^{n} a_i \phi_i^M$ will be smooth. Thus, if we use a truncated functional representation matrix $C_{0..n \times 0...n}$ then the image $C\mathbf{a}$ of any function f will be smooth.

Continuity of the representation. Finally, we also note that the functional representation is more amenable to numerical optimization since it is inherently continuous. That is, the matrix C can be modified continuously and still produce meaningful results. Note that there are no inherent restrictions on the matrix C to be able to establish functional correspondences. Thus, given *any* matrix C and any vector of coefficients $\mathbf{a}$, we can interpret $C\mathbf{a}$ as a functional mapping. To illustrate this, in Figure 5 we show the image of a set of three functions from a fixed source shape (shown in Figure 1) onto a target shape under a mapping matrix, interpolated between two mappings corresponding to the direct and symmetric shape matching. Note that each mapping is both meaningful and produces intuitive results.

5.3 Linearity of constraints

Perhaps even more importantly, the functional representation is particularly well suited for map inference (i.e. constrained optimization) for the following reason: when the underlying map T (and by extension the matrix C) are unknown, many natural constraints on the map become *linear* constraints in its functional representation. Below we describe the most common scenarios.

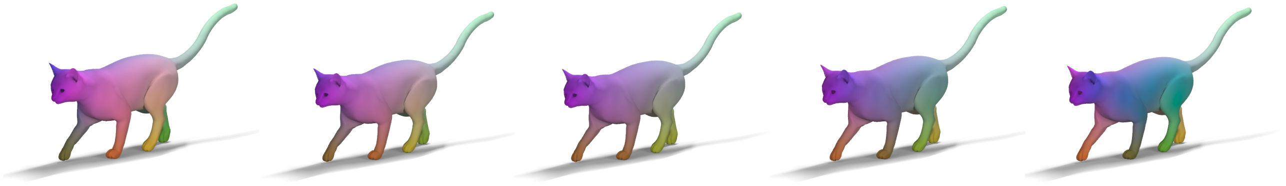
Function preservation. Given a pair of functions $f : M \to \mathbb{R}$ and $g : N \to \mathbb{R}$, the correspondence between f and g can be written simply as $C\mathbf{a} = \mathbf{b}$ where C is the functional representation of the map, while $\mathbf{a}$ and $\mathbf{b}$ are the representation of f and g in the chosen bases of M and N. Note that the function preservation constraint can be phrased entirely in terms of the matrix C without knowledge of the underlying correspondence T, since $\mathbf{a}$ and $\mathbf{b}$ do not depend on the map C. This is especially useful for shape matching applications where C is unknown, but could possibly be recovered by phrasing enough constraints of type $C\mathbf{a}_i = \mathbf{b}_i$. The function preservation constraint is quite general and includes the following as special cases.

Descriptor preservation. If f and g are functions corresponding to point descriptors (e.g. $f(x) = \kappa(x)$ where $\kappa(x)$ is Gauss curvature of M at x), then the function preservation constraint simply says that descriptors are approximately preserved by the mapping. Furthermore if the point descriptors are multidimensional so that $f(x) \in \mathbb{R}^k$ for each x then we can phrase k scalar function preservation constraints, one for each dimension of the descriptor.

Landmark point correspondences. If we are given landmark point correspondences $T(x) = y$ for some known $x \in M$ and $y \in N$ (e.g. specified by the user or obtained automatically), we can phrase this knowledge as functional constraints by considering functions f and g that are e.g. distance functions to the landmarks or normally distributed functions around x and y. Indeed, the confidence with which the landmark correspondence is known can be encoded in the functional constraints very naturally (e.g. if it is only known within a certain radius).

Segment correspondences. Similarly, if we are given correspondences between parts of shapes rather than individual points we can phrase such correspondences as functional correspondences again by either considering the indicator functions on the segments or more robust derived quantities such as the distance function.

In our implementation for finding functional maps between shapes,



(a) $\alpha = 0$ **(b)** $\alpha = 0.25$ **(c)** $\alpha = 0.5$ **(d)** $\alpha = 0.75$ **(e)** $\alpha = 1$

Figure 5: *Mapping of the three coordinate functions from the source shape shown in Figure 3a onto the target shape using an interpolation between two maps $C = \alpha C_1 + (1-\alpha)C_2$. Note that the mapped function varies continuously under changes of the parameter α.*

we impose a variety of functional constraints as described above. We will discuss the actual choice of the functions used below.

5.4 Operator Commutativity

In addition to the function preservation constraint, another class of constraints on the map that induce linear constraints on its functional representation is commutativity with respect to linear operators on M and N. That is, often M and N can be endowed with linear functional operators that we may want to preserve. A first example is a symmetry operator $S_F : \mathcal{F}(M,\mathbb{R}) \to \mathcal{F}(M,\mathbb{R})$ which associates with every function $f : M \to \mathbb{R}$ another function $S_F(f) : M \to \mathbb{R}$ obtained as $S_F(f)(x) = f(S^{-1}(x))$, where $S : M \to M$ is some symmetry of M. A second example is the Laplace-Beltrami operator and derived operators (e.g. the heat operator), which are preserved under isometries. The operators on M and N can be quite general, however, and can represent any association of functions on the manifold. In any case, given functional operators S_F and R_F on M and N respectively, it may be natural to require that the functional map C commute with these operators. In particular: $R_F C = C S_F$ or $\|R_F C - C S_F\| = 0$. This constraint, despite its second order flavor, leads to linear equations in the elements of C.

5.5 Regularization Constraints

Note that although we mentioned in Section 5.2 that there are no inherent constraints on the matrix C to be a functional map, this does not mean that *any* matrix C is associated with a point-to-point map. Indeed, while every bijective map T has a functional representation through the matrix C, the converse is not necessarily true. Thus, there may be constraints on the functional representation *if it is known to be associated with a point-to-point map*. Although finding such constraints seems to be a difficult open problem, a very useful observation is the following:

Theorem 5.1. *(1) If the basis functions are discrete and orthonormal with respect to the standard inner product, i.e. $\sum_x \phi_i(x)\phi_j(x) = \delta_{ij}$, or if the underlying map T (discrete or continuous) is volume preserving, i.e. $\mu^M(x) = \mu^N(T(x))$ where μ^M and μ^N are volume elements on M and N respectively, then the matrix C associated with the functional representation of T must be* orthonormal. *(2) If the underlying map T is an isometry then T commutes with the Laplace-Beltrami operator.*

Proof. See Appendix □

It follows that in most natural settings, e.g. when one expects isometries between shapes, if one is using the functional representation to obtain a point-to-point map it is most meaningful to consider orthonormal or nearly-orthonormal functional map matrices. Furthermore, it makes sense to incorporate commutation with the Laplace-Beltrami operators into the regularization.

5.6 Map Inversion and Composition

A challenging task when considering point-to-point mappings between shapes is map inversion, i.e. given a map $T : M \to N$ that is not necessarily bijective, one is required to find a meaningful version of $T^{-1} : N \to M$. In the functional representation finding an inverse can be done simply by finding an inverse of the mapping matrix C. Moreover, because for near-isometric maps we expect this matrix to be close to diagonal (or "funnel" shaped as shown in Figure 4) it is reasonable to take the inverse of the approximating submatrix of C. Finally, in light of Theorem 5.1 this can be done by simply taking the transpose of C or its approximation.

Similarly, map composition becomes simple matrix multiplication in the functional representation. We exploit these properties when we use our representation for joint map inference on a collection of shapes in Section 8.2.

6 Functional Map Inference

As mentioned in Section 5, functional shape maps are well-suited for inference because of their continuous nature and because a large number of constraints become linear in this representation. In this section we discuss how such inference can be done in practice. For this suppose we are given a pair of discrete shapes represented as meshes, with the corresponding Laplace-Beltrami eigenfunctions. Our goal is to find the underlying functional map represented as a matrix C. The simplest way to do so is to construct a large system of linear equations, where each equation corresponds to one of the constraints mentioned above (either a functional constraint or the operator commutativity constraint) and find the best functional map by finding the matrix C that best satisfies the constraints in the least squares sense.

6.1 Efficient Conversion to Point-to-Point

As mentioned in Section 4, given a bijection T between two discrete shapes, and the basis vectors of their function spaces, the functional representation C of the map T can be obtained by solving a linear system.

To reconstruct the original mapping from the functional representation, however, is more challenging. The simplest method alluded to in Remark 4.1 to find a corresponding point $y \in N$ to a given point $x \in M$ would require constructing a function $f : M \to \mathbb{R}$ (either the indicator function, or a highly peaked Gaussian around x) obtaining its image $g = T_F(f)$ using C and declaring y to be the point at which $g(y)$ obtains the maximum. Such a method, however, would require $O(V_N V_M)$ operations for a pair of meshes with V_N and V_M vertices. Such complexity may be prohibitively expensive in practice for large meshes. To obtain a more efficient method, note that in the Laplace-Beltrami basis δ_x, the delta function around a point $x \in M$, has the co-

Algorithm 1 FUNCTIONAL MAP INFERENCE FOR MATCHING

```
1. Compute a set of descriptors for each point on $M$ and $N$, and create function preservation constraints.
2. If landmark correspondences or part decomposition constraints are known, compute the function preservation constraints using those.
3. Include operator commutativity constraints for relevant linear operators on $M$ and $N$ (e.g. Laplace-Beltrami or symmetry).
4. Incorporate the constraints into a linear system and solve it in the least squares sense to compute the optimal $C$.
5. Refine the initial solution $C$ using the iterative method in Section 6.2.
6. If point correspondences are required, obtain them using the method in Section 6.1.
```

efficients: $a_i = \phi_i^M(x)$. This can be seen for example, since $\delta_x = \lim_{t \to 0^+} k_t^M(x, \cdot) = \lim_{t \to 0^+} \sum_{i=0}^{\infty} e^{-t\lambda_i} \phi_i^M(x) \phi_i^M(\cdot)$, where $k_t^M(\cdot, \cdot)$ is the heat kernel at time t on the shape M.

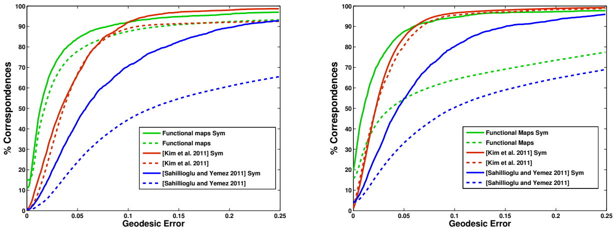
Therefore, given a matrix Φ^M of the Laplace-Beltrami eigenfunctions of M, where each column corresponds to a point and each row to an eigenfunction, one can find the image of *all* of the delta functions centered at points of M simply as $C\Phi^M$. Now recall that by Plancherel's theorem, given two functions g_1 and g_2 both defined on N, with spectral coefficients $\mathbf{b}_1$ and $\mathbf{b}_2$, $\sum_i (b_{1i} - b_{2i})^2 = \int_N (g_1(y) - g_2(y))^2 \mu(y)$. That is, the distances between the coefficient vectors is equal to the L^2 difference between the functions themselves. Therefore an efficient way to find correspondences between points is to consider for every point of $C\Phi^M$ its nearest neighbor in Φ^N. Using an efficient proximity search structure, such as a kd-tree, this procedure will require only $O(V_N \log V_N + V_M \log V_N)$ operations, giving a significant efficiency increase in practice.

6.2 Post-Processing Iterative Refinement

The observation made in Section 6.1 can also be used to refine a given matrix C to make it closer to a point-to-point map. Suppose we are given an initial estimate matrix C_0 that we believe comes from a point-to-point map T. As noted in Section 6.1, theoretically C_0 must be such that each column of $C_0\Phi^M$ coincides with some column of Φ^N. If we treat Φ^M and Φ^N as two point clouds with dimensionality equal to the number of eigenvalues used in the functional representation C_0 then this means that C_0 must *align* Φ^M and Φ^N. Moreover, since by Theorem 5.1 we expect the mapping matrix C_0 to be orthonormal, we can phrase the problem of finding the optimal mapping matrix C as rigid alignment between Φ^M and Φ^N. Thus an iterative refinement of C_0 can be obtained via:

1. For each column x of $C_0\Phi^M$ find the closest $\tilde{x}$ in Φ^N.
2. Find the optimal orthonormal C minimizing $\sum \|Cx - \tilde{x}\|$.
3. Set $C_0 = C$ and iterate until convergence.

Note that this technique is identical to the standard Iterative Closest Point algorithm of Besl & McKay, [1992], except that it is done in the embedded functional space, rather than the standard Euclidean space. Note also that this method *cannot* be used on its own to obtain the optimal functional matrix C because the embedding Φ^M and Φ^N are only defined up to a sign change (or more generally an orthonormal transformation within an eigenspace). Therefore, it is essential to have a good initial estimate matrix C_0. Finally, note that the output of this procedure is not only a functional matrix C but also a point-to-point correspondence given by nearest neighbor assignment between points on M and N. We will use this method to obtain good point-to-point maps when we apply these observations to devise an efficient shape matching method in Section 7.



(a) *SCAPE*　　**(b)** *TOSCA*

Figure 6: *Comparison of our method with the state-of-the-art methods of Kim et al. [2011] and Sahillioglu and Yemez [2011] on two datasets: SCAPE [Anguelov et al. 2005] and TOSCA [Bronstein et al. 2008] with and without symmetric maps allowed (solid and dashed lines respectively). Note that since our method is intrinsic only symmetric (solid line) evaluation is meaningful.*

Relation to Existing Methods. Note that this refinement step is similar to existing spectral matching methods such as [Jain et al. 2007; Mateus et al. 2008; Sharma and Horaud 2010]. However, in addition to having a good initial estimate C_0, our method is different since we allow "mixing" across eigenvectors corresponding to different eigenvalues. In addition, the functional representation allows to formulate other constraints such as operator commutativity (Section 5.4) and *represent* the map itself compactly.

7 Shape Matching

In this section we describe a simple yet very effective method for non-rigid shape matching based on the functional representation of mappings between shapes.

The simplest version of the shape matching algorithm is summarized in Algorithm 1. Namely, suppose we are given two shapes M and N in their discrete (e.g. mesh) representation, and the Laplace-Beltrami eigen-decomposition, we simply compute functional constraints that correspond to descriptor and segment preservation constraints together with the operator commutativity, form a linear system of equations and solve it in the least squares sense. If necessary, we refine the solution using the method in Section 6.2 and compute the point-to-point map using the method in Section 6.1.

7.1 Implementation

The key ingredients necessary to implement this method in practice are the computation of the eigendecomposition of the Laplace-Beltrami operator, the descriptors used in the function preservation constraints, and a method to obtain landmark or segment correspondences. Note that our framework allows great flexibility for the choice of descriptors and correspondence constraints since they all fit into a general function preservation framework. In our implementation we have used the cotangent scheme [Meyer et al. 2002] for the Laplace-Beltrami operator on meshed surfaces. We also used the Wave Kernel Signature (WKS) and Heat Kernel Signature descriptors of [Aubry et al. 2011] and [Sun et al. 2009]. Because the method described above is fully intrinsic and does not distinguish between left and right symmetries, it is also important to resolve ambiguities using correspondence constraints. However, since point-to-point correspondences (e.g. landmark) are generally unstable and difficult to obtain without manual intervention, we have used segment correspondences instead. Towards this goal, we first pre-segment every shape using the persistence-based segmentation technique of [Skraba et al. 2010] with the WKS at a fixed

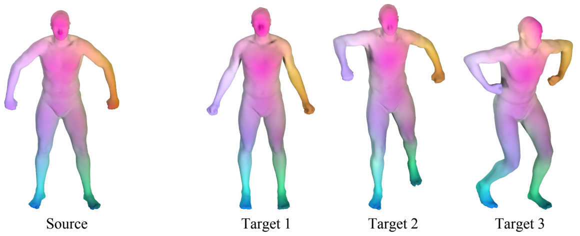


Figure 7: *Maps between remeshed versions of shapes from the SCAPE collection, mapping the coordinate functions from the source to the three target shapes using an inferred functional map.*

energy value of the underlying function (we used $e = 5$ in all examples below). This gives a relatively stable segmentation with a small number of segments (between 3 and 7 in the shapes we examined). Given a pair of shapes, we first compute the segment correspondence constraints. For this, we first compute the set of candidate pairs of segments from the two shapes by computing segment descriptors and finding the ones likely to match. For segment descriptors we use the sum of the WKS values of the points in the segment. Given a pair of candidate segment matches $s1, s2$ on M and N respectively, we construct a set of functional constraints using the Heat Kernel Map [Ovsjanikov et al. 2010] based on segment correspondences. We combine these together with the Laplace-Beltrami commutativity constraint and the WKS constraints into a single linear system and solve it to find the optimal functional mapping matrix C. Finally, we refine the solution using the iterative method described in Section 6.2 and find the final dense point-to-point correspondences using the method in 6.1.

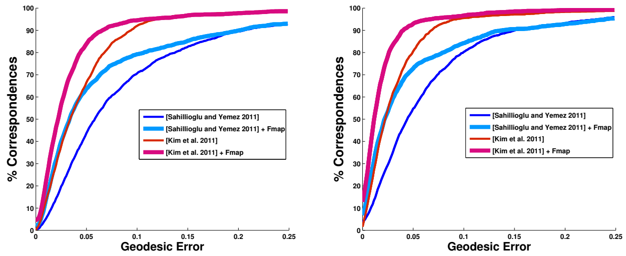
7.2 Results

We have evaluated our basic method for computing point-to-point correspondences on the shape matching benchmark of Kim et al. [2011] in which the authors showed state-of-the art results using their Blended Intrinsic Maps (BIM) approach. Using the correspondence evaluation, Figure 6 shows the results of our automated shape matching on two standard datasets used in the benchmark of Kim et al. [2011] on which their method reported significant improvement over prior work. In addition, we evaluated a recent shape matching method by Sahillioglu and Yemez [2011] which did not appear in the benchmark. The graphs show the percent of correspondences which have geodesic error smaller than a threshold. Note that our method shows quality improvement over Blended Intrinsic Maps on both datasets. Note also that all of the parameters for our system were fixed before running the benchmark evaluation and were therefore not optimized for benchmark performance in any way.

Although the shapes in both SCAPE and TOSCA datasets have the same connectivity structure, this information is not used by our method, and is not needed for applying our algorithm. To demonstrate this, Figure 7 shows three maps computed by our method between a source and three target shapes from the SCAPE collection, all remeshed with uniform remeshing. We show the map by transferring the XYZ coordinate functions to the target shapes using the inferred functional maps. These functions are then rendered as RGB channels on the source and target shapes.

8 Other Applications

8.1 Improving Point-to-Point Maps



(a) *SCAPE* **(b)** *TOSCA*

Figure 8: *Improvement of the results other methods using our representation and iterative refinement.*

Although the method presented above achieves state-of-the art performance on an isometric shape matching benchmark, its true strength lies perhaps not in the exact optimization decisions (e.g. the descriptors or segmentation techniques used) but rather in the representation of the mapping itself. To illustrate this, we have used the functional representation to improve the quality of other isometric shape matching methods. Here, instead of computing a new point-to-point map between a pair of shapes, we assume that we are given a point-to-point map obtained by any external method and our goal is to improve it through local refinement. The functional representation is a natural choice for this task because conversion to and from this representation is very fast (Sections 4 and 6.1) while iterative refinement does not require expensive non-linear optimization techniques. To test this approach, we used the point-to-point maps provided by the methods of Kim et al. [2011] and Sahillioglu and Yemez [2011], and converted them into a functional representation using the basis provided by the first 100 eigenfunctions of the cotangent weight scheme [Meyer et al. 2002] of the Laplace-Beltrami operator. We then ran the iterative refinement procedure described in Section 6.2 for 20 iterations, and converted the functional representation back to a point-to-point map (Section 6.1). For models ranging in 10-50 thousand points this entire procedure took 5-15 seconds on one core of Intel Xeon 3.2 GHz on average. Figure 8 shows the improvement in the quality of the final refined maps. Note that in both datasets and both sets of input maps this simple technique achieves significant improvements in the quality of the final maps.

Figure 9 illustrates the types of improvements provided by this iterative refinement method, For four sample pairs of meshes it shows the error of the initial mapping provided by Kim et al. [2011] and the error of the map after refinement. The color of each vertex is proportional to the distance of its match from the ground truth. As can be seen from this figure qualitatively, iterative refinement is able to recover from errors if the initial map is approximately correct.

8.2 Shape Collections

In addition to improving maps between pairs of shapes, our representation is also beneficial for improving collections of shape maps.

ICSM Recently it has been shown, [Nguyen et al. 2011], that when given a collection of shapes and maps between them, the pairwise maps can be considerably improved by considering the context provided by the collection. The main idea is to compose maps along cycles (e.g. compose the maps $L \to M$, $M \to N$, $N \to L$) in the resulting graph, and compare the composite map to the identity. When the composition result on a cycle is far from the identity, it is an indication that one or more of the maps on that cycle may be faulty. The paper [Nguyen et al. 2011] presented a method (which we will refer to as ICSM — Iteratively Corrected Shape Maps) to

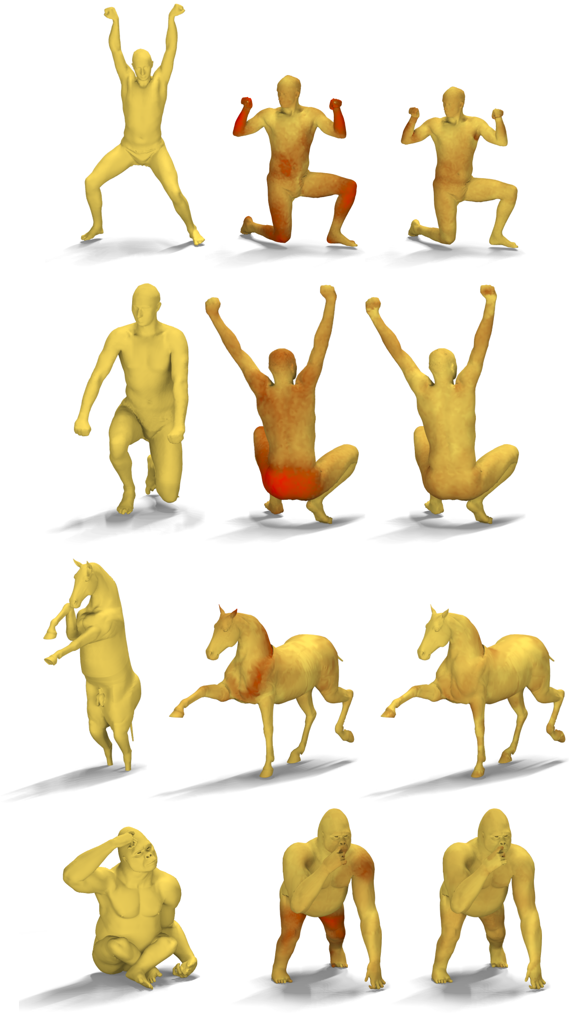
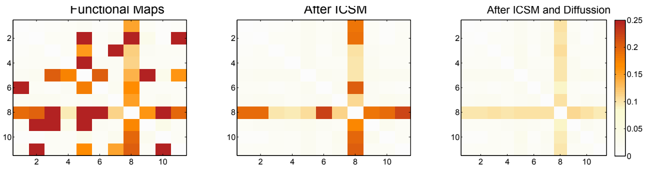


Figure 9: *Improvement of the maps provided by the method of Kim et al. [2011] using our functional representation with iterative refinement. For each pair of shapes, we show the source shape, and the errors in map of the target shape with color proportional to the distance from the ground truth. Note that iterative refinement efficiently removes spurious matches if the initial map is approximately correct.*

compute a score for the map between every pair of shapes, as a function of the consistency of the composed maps along 3-cycles. Then a shape graph is defined, where each shape is a vertex, and an edge between shapes is weighted by the computed score. Maps are improved by replacing the original maps with composed maps along the shortest paths in this graph.

ICSM can be easily applied using the functional maps representation, *without requiring conversion to point-to-point maps*. For map composition we simply use matrix multiplication, i.e. $m_{L,N} = m_{M,N} \circ m_{L,M} \equiv C_{MN}C_{LM}$, where C_{MN} and C_{LM} are the functional maps $M \to N$ and $L \to M$ respectively. Note, that if C_{MN} and C_{LM} are orthogonal matrices, the composition will also be orthogonal. As a measure of the distance of a functional map from the identity we use $\|C - I\|_{Fro}$ where I is the identity matrix.

Although it is possible to convert the functional maps to point-to-point maps and run ICSM on the point-to-point maps, it is advantageous to do it directly on the functional maps, as the evaluation of distance from identity is considerably less expensive: for point-to-point maps, one needs to compute the geodesic distance between the mapped point and the source point, whereas for functional maps we just need to compute the Frobenius norm of a matrix. Furthermore, evaluating the error of a point-to-point map will be more costly for a denser mesh, whereas the complexity remains the same for the functional map.



(a) **(b)** **(c)**

Figure 10: *(a) Average geodesic errors of maps on a collection of 11 shapes from the SCAPE benchmark. (b) Errors after running ICSM. (c) Errors after running diffusion on ICSM results.*

To evaluate the use of ICSM with functional maps, we generated all the maps between the first 11 meshes in the SCAPE database. As our method is fully intrinsic, some of the generated maps were symmetric (i.e. left side mapped to right side), and some were straight. Figure 10(a) shows the average geodesic error for all the pairs in the collection. The pairs which have large errors are usually pairs for which the symmetric map was chosen. Figure 10(b) shows the average geodesic error after applying ICSM to the functional maps, as described earlier. Note that the maps improved considerably, leaving only a single shape with noticeable errors.

Map diffusion In [Nguyen et al. 2011] map improvement was achieved by re-writing maps by compositions of other maps along shortest paths in the shape graph. However, if we use the functional map representation, we can define a more subtle map improvement, which takes into consideration additional information. Specifically, we can replace a map with a *weighted average* of other maps, i.e.: $\hat{C}_{ij} = \sum_k w_{kj}C_{kj}w_{ik}C_{ik}$, where C_{ij} is the map between the shapes M_i and M_j, and w_{ij} is a weight which describes our confidence in the map — thus being better able to deal with the effects of noise.

In fact, we can construct a "SuperMap" for the whole collection, by creating a matrix whose (i, j)-th block is $w_{ji}C_{ji}$. Computing the weighted average of maps described previously is equivalent to raising the "SuperMap" matrix to the power of 2. This approach can be thought of as applying heat diffusion in the space of maps, similarly to what has been done recently in [Singer and Wu 2011]. Our succinct map representation may allow for a generalization of [Singer and Wu 2011], from rotations in $\mathbb{R}^3$ to general maps between shapes.

We have applied this simple idea to the maps generated by ICSM, computing w_{ij} from the ICSM weights, and subsequently applying the ICP post-processing described in Section 6.2. Figure 10(c) shows the average geodesic errors of the resulting maps. As can be seen in the figure, map diffusion succeeded to further reduce the error on the problematic shape where ICSM could not improve further. Figure 11 shows the percent of correspondences which have geodesic error smaller than a threshold, for the maps generated by our matching method, the maps after ICSM, the maps after ICSM and diffusion, and the maps generated by [Kim et al. 2011]. Again, we can see that applying map diffusion to the result of ICSM improves the map collection.

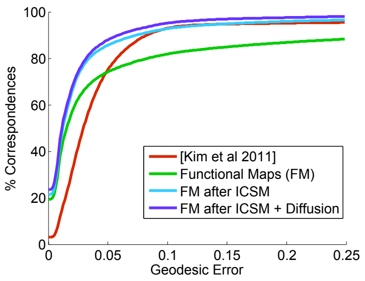


Figure 11: *Improving a collection of shape maps using ICSM and diffusion of maps. See the text for details.*

8.3 Function (Segmentation) Transfer

As mentioned earlier, one of the advantages of the functional representation is that it reduces the transfer of functions across shapes to a matrix product, without resorting to establishing point-to-point correspondences. This is particularly useful since function transfer is one of the key applications of maps between shapes and obtaining point-to-point correspondences is often challenging. We illustrate the performance of this idea on the task of segmentation transfer across shapes. Here we are given a pair of shapes where one of the shapes is pre-segmented and the goal is to find a compatible segmentation of the second shape. To achieve this task we simply construct an indicator function for each of the segments on the source shape and use the functional map to transfer this indicator function. Then each point on the target map will have a set of values for each of the transferred segments. Finally, if "hard" clustering is required, we associate with the point the segment that produced the maximum of these values.

Figure 12 shows this idea applied to several shapes from the TOSCA and SCAPE datasets. For each pair of shapes we show the image of the the indicator function of one of the segments as well as the final "hard" segmentation. Note that throughout this procedure no point-to-point correspondences were used.

9 Conclusion & Future Work

In this paper we have introduced a novel representation for maps between pairs of shapes that generalizes the standard notion of a map as a pairing of points. By concentrating on finding correspondences between generic functions defined on the shapes, we devised a representation that is multiscale, compact, and amenable to efficient optimization. Perhaps the most important property of the functional representation is that many natural constraints on a map become linear, making direct optimization feasible. We have demonstrated the effectiveness of our representation both by achieving state-of-the-art results on an isometric shape matching benchmark and by showing that other existing methods can benefit from this representation as a post-processing step. Finally, we have shown its effectiveness in segmentation transfer across pairs of shapes without establishing correspondences and on joint analysis of large collections of shapes.

In the future we will look more carefully at the optimal choice of bases for the functional representation, thus extending the scope of functional maps to more general classes of deformations.

Acknowledgements: The authors would like to acknowledge NSF grants FODAVA 0808515, IIS 0914833 and CCF 1011228, a research grant from Google, Inc., and the support of the Max Planck Center for Visual Computing and Communications.

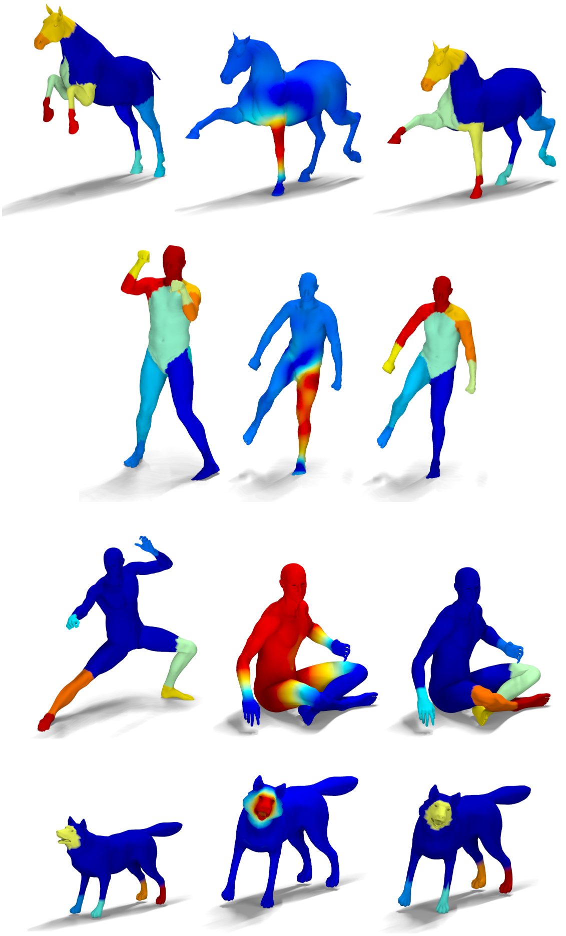


Figure 12: *Segmentation transfer using the functional map representation. For each pair of shapes we show 3 figures: the user-provided source segmentation of the first shape, the image of one of the indicator functions of a segment using the functional map computed with our method, and the final segmentation transfer onto the target shape. Note that point correspondences were not computed at any point during this procedure.*

References

Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., and Davis, J. 2005. SCAPE: Shape completion and animation of people. In *Proc. SIGGRAPH*, 408–416.

Aubry, M., Schlickewei, U., and Cremers, D. 2011. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Proc. ICCV - 4DMOD Workshop*.

Besl, P. J., and McKay, N. D. 1992. A method for registration of 3-d shapes. *IEEE TPAMI 14*, 239–256.

Bronstein, A. M., Bronstein, M. M., and Kimmel, R. 2006. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *PNAS 103*, 5, 1168–1172.

Bronstein, A., Bronstein, M., and Kimmel, R. 2008. *Numerical Geometry of Non-Rigid Shapes*. Springer.

Çela, E. 1998. *The Quadratic Assignment Problem: Theory and Algorithms*. Kluwer Academic Publishers.

Dubrovina, A., and Kimmel, R. 2011. Approximately isometric shape correspondence by matching pointwise spectral features and global geodesic structures. *Advances in Adaptive Data Analysis*, 203–228.

Golovinskiy, A., and Funkhouser, T. 2009. Consistent segmentation of 3D models. *Computers and Graphics (Proc. SMI) 33*, 3, 262–269.

Huang, Q.-X., Adams, B., Wicke, M., and Guibas, L. J. 2008. Non-rigid registration under isometric deformations. *CGF (Proc. SGP) 27*, 5, 1449–1457.

Huang, Q., Koltun, V., and Guibas, L. 2011. Joint shape segmentation with linear programming. *ACM TOG (Proc. SIGGRAPH Asia) 30*, 6, 125:1–125:12.

Jain, V., Zhang, H., and van Kaick, O. 2007. Non-rigid spectral correspondence of triangle meshes. *International Journal on Shape Modeling 13*, 1, 101–124.

Kalogerakis, E., Hertzmann, A., and Singh, K. 2010. Learning 3D Mesh Segmentation and Labeling. *ACM Transactions on Graphics 29*, 3.

Kato, T. 1995. *Perturbation Theory for Linear Operators.* Springer-Verlag GmbH.

Kim, V. G., Lipman, Y., and Funkhouser, T. 2011. Blended intrinsic maps. In *Proc. SIGGRAPH*, 79:1–79:12.

Kin-Chung Au, O., Tai, C.-L., Cohen-Or, D., Zheng, Y., and Fu, H. 2010. Electors voting for fast automatic shape correspondence. *Computer Graphics Forum 29*, 2, 645–654.

Lipman, Y., and Funkhouser, T. 2009. Möbius voting for surface correspondence. In *Proc. SIGGRAPH*, 72:1–72:12.

Mateus, D., Horaud, R. P., Knossow, D., Cuzzolin, F., and Boyer, E. 2008. Articulated shape matching using Laplacian eigenfunctions and unsupervised point registration. In *Proc. CVPR*, 1–8.

Meyer, M., Desbrun, M., Schröder, P., and Barr, A. H. 2002. Discrete differential-geometry operators for triangulated 2-manifolds. In *Proc. VisMath*, 35–57.

Nguyen, A., Ben-Chen, M., Welnicka, K., Ye, Y., and Guibas, L. 2011. An optimization approach to improving collections of shape maps. In *Proc. SGP*, 1481–1491.

Ovsjanikov, M., Sun, J., and Guibas, L. 2008. Global intrinsic symmetries of shapes. *Comp. Graph. Forum 27*, 5, 1341–1348.

Ovsjanikov, M., Merigot, Q., Memoli, F., and Guibas, L. 2010. One point isometric matching with the heat kernel. *CGF 29*, 5, 1555–1564.

Pokrass, J., Bronstein, A. M., and Bronstein, M. M. 2011. A correspondence-less approach to matching of deformable shapes. In *SSVM*, 592–603.

Rustamov, R. M. 2007. Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Proc. SGP*, 225–233.

Sahillioğlu, Y., and Yemez, Y. 2011. Coarse-to-fine combinatorial matching for dense isometric shape correspondence. *Computer Graphics Forum 30*, 5, 1461–1470.

Sharma, A., and Horaud, R. P. 2010. Shape matching based on diffusion embedding and on mutual isometric consistency. In *Proc. CVPR – NORDIA Workshop*, 29–36.

Singer, A., and Wu, H. 2011. Vector diffusion maps and the connection laplacian. *Arxiv preprint arXiv:1102.0075*.

Skraba, P., Ovsjanikov, M., Chazal, F., and Guibas, L. 2010. Persistence-based segmentation of deformable shapes. In *Proc. CVPR – NORDIA Workshop*, 45–52.

Sun, J., Ovsjanikov, M., and Guibas, L. 2009. A concise and provably informative multi-scale signature based on heat diffusion. In *Proc. SGP*, 1383–1392.

Tevs, A., Berner, A., Wand, M., Ihrke, I., and Seidel, H.-P. 2011. Intrinsic shape matching by planned landmark sampling. *CGF (Proc. Eurographics) 30*, 2, 543–552.

van Kaick, O., Tagliasacchi, A., Sidi, O., Zhang, H., Cohen-Or, D., Wolf, L., and Hamarneh, G. 2011. Prior knowledge for part correspondence. *CGF (Proc. Eurographics) 30*, 2, 553–562.

van Kaick, O., Zhang, H., Hamarneh, G., and Cohen-Or, D. 2011. A survey on shape correspondence. *Computer Graphics Forum 30*, 6, 1681–1707.

Weyl, H. 1946. *The Classical Groups: Their Invariants and Representations*. Princeton University Press.

Xu, K., Li, H., Zhang, H., Cohen-Or, D., Xiong, Y., and Cheng, Z. 2010. Style-content separation by anisotropic part scales. *ACM TOG (Proc. SIGGRAPH Asia) 29*, 5, 184:1–184:10.

Yeh, I.-C., Lin, C.-H., Sorkine, O., and Lee, T.-Y. 2010. Template-based 3d model fitting using dual-domain relaxation. *IEEE Transactions on Visualization and Computer Graphics 99*.

Zhang, H., Sheffer, A., Cohen-Or, Zhou, Q., van Kaick, O., and Tagliasacchi, A. 2008. Deformation-driven shape correspondence. In *Proc. SGP*, 1431–1439.

A Proof of Theorem 5.1

It is well-known that isometries commute with the Laplace-Beltrami operator thus we need only concentrate on the proving the first statement. Therefore suppose that the map T is volume-preserving. This means, in particular, that $\mu^N(T(x)) = \mu^M(x) \ \forall \ x \in M$, where μ is the volume element. Recall that in the LB basis: $C_{i,j} = \left\langle T(\phi_i^M), \phi_j^N \right\rangle = \int_N \phi_i^M \circ T^{-1}(y)\phi^N(y)\mu^N(y)$. Now consider the map $S = T^{-1} : N \to M$ and let D be its functional representation. Then $D_{j,i} = \left\langle S(\phi_j^N), \phi_i^M \right\rangle = \int_M \phi_j^N \circ T(x)\phi^M(x)\mu^M(x)$. Note that the integrands in $C_{i,j}$ and $D_{j,i}$ are the same. Moreover, since T is volume preserving, the integrals themselves have to agree. It follows that $C = D^T$. But since D is the functional representation of the inverse map T^{-1} we must have $CD = DC = Id$. Therefore, $C^T C = Id$ and C is orthonormal. The proof under the assumption of discrete orthonormality is identical.